

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-230844

(43)公開日 平成6年(1994)8月19日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 1/00

9/06

識別記号

3 7 0 B 7165-5B

4 1 0 B 9367-5B

庁内整理番号

F I

技術表示箇所

審査請求 有 請求項の数 5 O L (全 9 頁)

(21)出願番号

特願平5-16677

(22)出願日

平成5年(1993)2月4日

(71)出願人 591128763

株式会社富士通ソーシャルサイエンスラボ  
ラトリ

東京都品川区大崎1丁目6番4号

(72)発明者 望月 申一

東京都品川区大崎1丁目6番4号 株式会  
社富士通ソーシャルサイエンスラボラトリ  
内

(74)代理人 弁理士 岡田 守弘

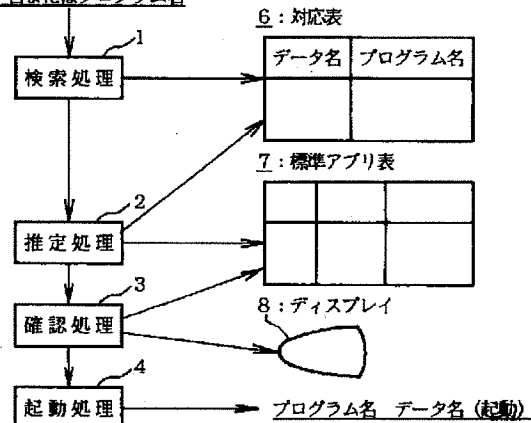
(54)【発明の名称】 処理起動方法

(57)【要約】

【目的】 本発明は、処理を起動する処理起動方法に関し、対応表および標準アプリ表を設け、データ名の指定に対応して検索してプログラム名や推定したプログラム名を表示してプログラムを起動し、データ名を指定するのみで簡単にプログラムを起動することを目的とする。

【構成】 データ名とプログラム名を対応づけて登録する対応表6を備え、データ名の指定に対応して、対応表6を検索して対応するプログラム名を表示し、実行指示に対応して当該表示したプログラム名と指定されたデータ名とによってプログラムを起動するように構成する。

本発明の原理構成図  
データ名またはプログラム名



## 【特許請求の範囲】

【請求項 1】 処理を起動する処理起動方法において、データ名とプログラム名を対応づけて登録する対応表 (6) を備え、

データ名の指定に対応して、上記対応表 (6) を検索して対応するプログラム名を表示し、実行指示に対応して当該表示したプログラム名と指定されたデータ名とによってプログラムを起動するように構成したことを特徴とする処理起動方法。

【請求項 2】 処理を起動する処理起動方法において、データ名とプログラム名を対応づけて登録する対応表 (6) と、

データの種類に対応づけてプログラム名を登録する標準アプリ表 (7) とを備え、

データ名の指定に対応して、上記対応表 (6) を検索して対応するプログラム名が見つかったときにそのプログラム名を表示し、一方、見つからなかったときに、上記標準アプリ表 (7) を検索して当該指定されたデータ名のデータの種類の種類に対応するプログラム名を取り出して表示し、実行指示に対応して当該プログラム名と指定されたデータ名とによってプログラムを起動するように構成したことを特徴とする処理起動方法。

【請求項 3】 処理を起動する処理起動方法において、データ名とプログラム名を対応づけて登録する対応表 (6) と、

データの特徴に対応づけてプログラム名を登録する標準アプリ表 (7) とを備え、

データ名の指定に対応して、上記対応表 (6) を検索して対応するプログラム名が見つかったときにそのプログラム名を表示し、一方、見つからなかったときに、上記標準アプリ表 (7) を検索して当該指定されたデータ名のデータの特徴に対応するプログラム名を取り出して表示し、実行指示に対応して当該プログラム名と指定されたデータ名とによってプログラムを起動するように構成したことを特徴とする処理起動方法。

【請求項 4】 上記プログラム名の表示に対応して、他のプログラム名が入力されて実行指示された場合に、指定されたデータ名および実行指示された他のプログラム名を対応づけて上記対応表 (6) に登録するように構成したことを特徴とする請求項 1 記載から請求項 3 記載の処理起動方法。

【請求項 5】 上記標準アプリ表 (7) を検索して当該指定されたデータ名のデータの種類の種類および特徴のいずれも見つからなかったとき (あるいは必要に応じて)、当該データ名のデータの種類の種類およびデータの特徴の少なくとも 1 つの入力に対応して、上記標準アプリ表 (7) にプログラム名に対応づけて登録するように構成したことを特徴とする請求項 3 記載および請求項 4 記載の処理起動方法。

【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、処理を起動する処理起動方法に関するものである。

## 【0002】

【従来の技術】 従来、データを指定して、これに対応するプログラムを起動する場合、以下のいずれかの方法を採用していた。

【0003】 (1) プログラムの情報をデータの中へ埋め込んでおく。これにより、利用者がデータを指定すると、その中に埋め込まれているプログラム情報をもとに対応するプログラムを起動する。

【0004】 (2) データの名前にそれを利用するプログラム情報を付加する。これにより、利用者がデータを指定すると、データの名前に付加されているプログラム情報を取り出してこれに対応するプログラムを起動する。

## 【0005】

【発明が解決しようとする課題】 上述した従来の (1) の方法によればプログラムの情報をデータ中に埋め込む煩雑性があり、(2) の方法によればプログラムの情報をデータの名前に付加する煩雑性があり、いずれも利用者にデータの種類の種類を気にしたり、操作が煩雑となってしまうたり、簡単な操作によってプログラムを起動し得ないという問題があった。

【0006】 本発明は、これらの問題を解決するため、対応表および標準アプリ表を設け、データ名の指定に対応してこれらを検索して対応するプログラム名や推定したプログラム名を表示してプログラムを起動し、データ名を指定するのみで簡単にプログラムを起動することを目的としている。

## 【0007】

【課題を解決するための手段】 図 1 は、本発明の原理構成図を示す。図 1 において、対応表 6 は、データ名とプログラム名を対応づけて登録するものである。

【0008】 標準アプリ表 7 は、データの種類の種類およびデータの特徴に対応づけてプログラム名を登録するものである。

## 【0009】

【作用】 本発明は、図 1 に示すように、データ名の指定に対応して、対応表 6 を検索して対応するプログラム名を表示し、実行指示に対応して当該表示したプログラム名と指定されたデータ名とによってプログラムを起動するようにしている。

【0010】 また、データ名の指定に対応して、対応表 6 を検索して対応するプログラム名が見つかったときにそのプログラム名を表示し、一方、見つからなかったときに、標準アプリ表 7 を検索して当該指定されたデータ名のデータの種類の種類に対応するプログラム名を取り出して表示し、実行指示に対応して当該プログラム名と指定されたデータ名とによってプログラムを起動するようにし

ている。

【0011】また、データ名の指定に対応して、対応表6を検索して対応するプログラム名が見つかったときにそのプログラム名を表示し、一方、見つからなかったときに、標準アプリ表7を検索して当該指定されたデータ名のデータの特徴に対応するプログラム名を取り出して表示し、実行指示に対応して当該プログラム名と指定されたデータ名とによってプログラムを起動するようにしている。

【0012】この際、プログラム名の表示に対応して、他のプログラム名が入力されて実行指示された場合に、指定されたデータ名および実行指示された他のプログラム名を対応づけて対応表6に登録するようにしている。

【0013】また、標準アプリ表7を検索して当該指定されたデータ名のデータの種類の種類および特徴のいずれも見つからなかったとき（あるいは必要に応じて）、当該データ名のデータの種類の種類およびデータの特徴の少なくとも1つの入力に対応して、標準アプリ表7にプログラム名に対応づけて登録するようにしている。

【0014】従って、対応表6および標準アプリ表7を設け、データ名の指定に対応してこれらを検索して対応するプログラム名や推定したプログラム名を表示し、実行指示に対応してプログラムを起動することにより、データ名を指定するのみで簡単にプログラムを起動することが可能となる。

【0015】

【実施例】次に、図1から図7を用いて本発明の実施例の構成および動作を順次詳細に説明する。

【0016】図1は、本発明の原理構成図を示す。図1において、検索処理1は、指定されたデータ名あるいはプログラム名をもとに、対応表6を検索して対応するプログラム名あるいはデータ名を見ついたりなどするものである。

【0017】推定処理2は、検索処理1が対応表6を検索しても対応するプログラム名あるいはデータ名を見つけないことができなかった場合に、標準アプリ表7を検索して、データ名のデータの種類の種類（例えばテキスト、図形など）に対応するプログラム名を見ついたり、データ名のデータの特徴（例えば特定の命令コードがあるオブジェクトモジュール、ロードモジュールなど）に対応するプログラム名を見ついたりなどするものである。

データの種類の種類	プログラム名
テキスト	ワープロB
図形	ICAD
データベース	DB2

このように登録することにより、例えば利用者がテキスト名を指定したときに、対応表6にそのテキスト名が見つからなかったときに、当該テキストの種類（あるいはテキストの特徴）に対応するプログラムBを取り出して推定して表示することができる。この推定して表示した

\* 【0018】確認処理3は、検索処理1あるいは推定処理2によって見つけたデータ名に対応するプログラム名やプログラム名に対応するデータ名をディスプレイ8上に表示したり、この表示に対応して利用者からのOKの確認入力を受け取って起動処理4の実行を開始させたりなどするものである。

【0019】起動処理4は、利用者が確認して実行指示したデータ名およびプログラム名をもとにプログラムを起動するものである。対応表6は、データ名とプログラム名を対応づけて登録するものである。例えばデータ名（例えば文書A）およびプログラム名（ワープロB）と登録する（図2参照）。

【0020】標準アプリ表7は、データの種類の種類およびデータの特徴に対応づけてプログラム名を登録するものである。例えばデータの種類の種類（例えばテキスト）、プログラム名（例えばワープロB）、データの特徴（例えば漢字コードが多い）を登録する（図3参照）。

【0021】図2は、本発明の対応表例を示す。この対応表6は、データ名に対応づけてプログラム名を登録したものであって、例えば図示の下記のように登録する。

データ名	プログラム名
文書A	ワープロB
文書C	ワープロB

このように登録することにより、利用者が文書Aを指定したときに、この対応表6を検索してワープロBというプログラム名を取り出し、これらデータ名“文書A”およびプログラム名“ワープロB”をもとにプログラムを起動して文書Aの処理（編集）を行う。この際、対応表6に登録されていなかった場合には、図3の標準アプリ表7を検索して推定を行って表示し、利用者からの実行指示があったときにこのときのデータ名およびプログラム名を当該対応表6に登録する。これにより、学習を行い、対応表6には自動的に推定して確認した結果を順次登録する。また、対応表6に登録されていなかった場合、画面上から入力されたプログラム名を指定されたデータ名に対応づけて登録するようにしてもよい。

【0022】図3は、本発明の標準アプリ表例を示す。この標準アプリ表7は、データの種類の種類、データの特徴、およびプログラム名を対応づけて登録したものであって、例えば図示の下記のように登録する。

データの特徴
かな・漢字コードが多い
特定のヘッダなど

プログラム名を、利用者が見て適切と判断して実行指示したときは、上述した対応表6にテキスト名およびプログラム名を登録する。一方、利用者が見て適切でないと判断し、当該テキスト名（データ名）に対応する新たなプログラム名の入力を持って標準アプリ表7に登録（あ

るいは修正)する。

【0024】次に、図4のフローチャートに示す順序に従い、図1から図3の構成の動作を詳細に説明する。図4において、S1は、データ名を指定する。例えば利用者が“文書A”を指定する。

【0025】S2は、対応表を検索する。これは、図1の検索処理1が対応表6から、S1で指定されたデータ名(例えば“文書A”)に対応するプログラム名を検索する。

【0026】S3は、S2の検索して該当するプログラム名があるか判別する。YESの場合には、S1で指定されたデータ名に対応するプログラム名が対応表6を検索して見つかったので、S9で起動し、S10でデータを処理する。一方、NOの場合には、対応表6を検索して見つからなかったので、S4に進む。

【0027】S4は、プログラムの推定を行う。これは、後述する図5のフローチャートに従い、標準アプリ表7を検索し、S1で指定されたデータ名のデータの種類あるいはデータの特徴に対応するプログラム名を見つけて推定する。

【0028】S5は、S4で見つけて推定したプログラム名のメッセージを表示する。例えば右側に記載したように、

・プログラムはこれでよいですか。

【0029】・入力したものはプログラムです。実行してよろしいでしょうか。

S6は、S5のメッセージ表示に対応して、利用者がよいと入力したか判別する。YESの場合には、推定して確認したデータ名とプログラム名をS8で対応表6に登録し、S9に進む。一方、NOの場合には、推定したプログラム名が適切でない、あるいは推定できなかったもので、S7で利用者にプログラム名の入力を促す。そして、S8でこの入力されたプログラム名と指定されたデータ名を対応表6に登録し、S9に進む。

【0030】S9は、プログラムを起動する。これは、S1で指定されたデータ名およびプログラム名(S2で対応表6を検索して取り出したプログラム名、S4で標準アプリ表7を検索して見つけて推定・確認したプログラム名、あるいはS7で利用者が入力したプログラム名)をもとに、プログラムを起動する。

【0031】S10は、データを処理する。これは、例えば文書AをワープロBで編集する。以上によって、データ名の指定に対応して、対応表6を検索して該当するプログラム名が見つかったときはこのプログラムを起動してデータを処理する。対応表6を検索して見つからなかったときは、標準アプリ表7を検索してデータ名のデータの種類あるいはデータの特徴に対応するプログラム名を取り出して推定して表示し、利用者が確認してOKのときにデータ名およびプログラム名を対応表6に登録すると共にこのプログラムを起動してデータを処理す

る。利用者がNGあるいは標準アプリ表7から見つからなかったとき、プログラム名の入力に対応してデータ名およびプログラム名を対応表6に登録すると共にこのプログラムを起動してデータを処理する。これらの構成および学習機能により、利用者はデータ名を指定するという簡単な操作によってプログラムを容易に起動することが可能となる。

【0032】図5は、本発明の推定説明フローチャート(その1)を示す。これは、図4のS4のプログラムの推定の詳細フローチャートである。図5において、S11は、指定データの内容がテキストデータか判別する。これは、指定されたデータ名が対応表6に登録されていなかったもので、当該指定データ名のデータ(指定データ)の内容がテキストデータか判別する。ここで、テキストデータは、

- ・改行文字の間隔が例えば300バイト以内で、
- ・かつ0～255までの値(改行、画面クリアなど)が少なく
- ・かつ“a”から“z”までと80以上の値(漢字コード)が多い

という性質を持つので、当該性質が指定データの内容に存在するか判別する。YESの場合には、S18でテキストデータと推定し、S19で標準アプリ表7を検索して、該当するプログラム名を取り出し、推定を終了する。一方、S11のNOの場合には、S12に進む。

【0033】S12は、指定データがロードモジュールまたはオブジェクトモジュールに対応するか判別する。ここで、ロードモジュールまたはオブジェクトモジュールは、

- ・ロード、ストア命令に対応する値(コード)が多い
- ・ファイル構造が特定の形に決まっている

という性質を持つので、当該性質が指定データの内容に存在するか判別する。YESの場合には、S20でロードモジュールまたはオブジェクトモジュールと推定し、終了する。一方、S12のNOの場合には、S13に進む。

【0034】S13は、図形データか判別する。ここで、図形データは、

- ・特定コード
- ・TIFFの識別コード

を持つという性質があるので、当該性質が指定データの内容に存在するか判別する。YESの場合には、S21で図形データと推定し、S19で標準アプリ表7を検索して、該当するプログラム名を取り出し、推定を終了する。一方、S13のNOの場合には、S14に進む。

【0035】S14は、データの特徴があるか判別する。ここで、データの特徴は、例えば指定データに特定のヘッダなどが存在するという特徴であって、当該データの特徴が存在するか判別する。YESの場合には、S22で標準アプリ表7のプログラム名を取り出し、推定

を終了する。一方、NOの場合には、S15に進む。

【0036】S15は、データの種類の入力を促す。S16は、利用者がデータの種類の入力を行う。S17は、標準アプリ表へ登録する。これらS15からS17は、S11からS14のいずれにもNOであって、標準アプリ表7から検索できないと判明したので、データの種類の入力を促すメッセージに対応して、利用者に指定データの種類（例えばテキスト、図形、データベースなど）を入力させると共に、指定データを処理するプログラム名を入力させ、データの種類とプログラム名を標準アプリ表7に登録し、次回に備える。

【0037】以上によって、データ名の指定に対応して、対応表6に存在しないときに、標準アプリ表7を検索し、指定されたデータ名のデータの種類（テキスト、図形など）、データの特徴（特定のヘッダなど）に対応するプログラム名を取り出して推定する。データの種類の、データの特徴がなかったときは、該当するものを標準アプリ表7に、プログラム名に対応づけて登録する。これにより、指定されたデータ名に対応するプログラム名を当該データ名のデータの種類や特徴によって推定することが可能となると共に学習することが可能となる。

【0038】図6は、本発明の推定説明フローチャート（その2）を示す。これは、データの特徴をもとに標準アプリ表7を検索するときの手順である。図6において、S31は、標準アプリ表を検索する。これは、指定されたデータ名のデータの特徴について、標準アプリ表7を検索する。

【0039】S32は、データの特徴があるか判別する。YESの場合には、データの特徴が標準アプリ表7にあったので、S36で標準アプリ表7の該当するプログラム名と推定する。一方、NOの場合には、データの特徴が標準アプリ表7になかったので、S33に進む。

【0040】S33は、データの種類とプログラムの特徴の入力を促す。S34は、上記データを利用者が入力する。S35は、標準アプリ表へ登録する。これらS33からS35は、S32のNOで指定されたデータ名のデータの特徴が標準アプリ表7に登録されていなかったため、データの種類の入力を利用者に促したことに伴って、利用者がデータの種類の入力を行い、これらデータの種類の入力、更に起動するプログラム名を対応づけて標準アプリ表7へ登録する。

【0041】以上によって、指定されたデータ名のデータの特徴が標準アプリ表7に登録されていたときに該当するプログラム名と推定し、登録されていなかったときにデータの種類の、データの特徴およびプログラム名の入力を利用者に促し、入力されたものを対応づけて標準アプリ表7に登録する。これにより、次回からデータの種類の入力およびデータの特徴に対応するプログラム名が標準アプリ表7に登録されているので、即時に推定することが

可能となる。

【0042】図7は、本発明のシステム構成説明図を示す。これは、図1の構成の本システム12を、電子計算機のオペレーティングシステムと組み合わせたものである。ここで、本システム12は、図示のように既存のオペレーティングシステム13の入力側に設けてもよいし、オペレーティングシステム13の一部としてもよい。いずれの場合も、下段に記載したように、利用者からは点線内がオペレーティングシステムに見え、本システム12を意識することなく、データ名、プログラム名、あるいはプログラム名+データ名のいずれかの指定方式に従い、プログラムを起動してデータの処理を実行させることが可能となる。

【0043】図7において、利用者11は、指定方式として、

- ・データ名
- ・プログラム名
- ・プログラム名+データ名

のいずれかを、指定して本システム12に入力し、プログラムを起動してデータの処理を指示するものである。

【0044】本システム12は、図1の構成を持つものであって、データ名、プログラム名、プログラム名+データ名のいずれかの指定に対しても、オペレーティングシステム13にプログラム名+データ名によって起動依頼するものである。

【0045】オペレーティングシステム13は、電子計算機の全体を統括制御するシステムであって、ここでは、プログラム名+データ名を指定した起動依頼に対応して、プログラムを起動してデータを処理（例えば既述したワープロBを起動して文書Aを編集）させるものである。

【0046】

【発明の効果】以上説明したように、本発明によれば、データ名の指定に対応して、対応表6を検索して対応するプログラム名が見つかったときにそのプログラム名を表示し、一方、見つからなかったときに、標準アプリ表7を検索して当該指定されたデータ名のデータの種類の（あるいはデータの特徴）に対応するプログラム名を取り出して推定して表示し、実行指示に対応して当該プログラム名と指定されたデータ名をもとにプログラムを起動したり、推定したプログラム名と指定したデータ名を対応表6に登録したり、データの種類の入力およびプログラム名を標準アプリ表7に登録したりする構成を採用しているため、データ名を指定するのみで簡単にプログラムを起動することができる共に、対応表6および標準アプリ表7に学習機能を持たせて実行指示されたデータ名、プログラム名、データの種類の、データの特徴を登録することができる。これらにより、

(1) オペレーティングシステムに特別な機能がなくても本システムを付加し、データを指定するのみで対応

するプログラムを起動して処理を実行させることができる。

【0047】(2) 本発明を導入する前から存在するデータ名についても推定処理2により、データを指定するのみで対応するプログラムを起動して処理を実行させることができる。

【図面の簡単な説明】

【図1】本発明の原理構成図である。

【図2】本発明の対応表例である。

【図3】本発明の標準アプリ表例である。

【図4】本発明の動作説明フローチャートである。

【図5】本発明の推定説明フローチャート(その1)で\*

\*ある。

【図6】本発明の推定説明フローチャート(その2)である。

【図7】本発明のシステム構成説明図である。

【符号の説明】

1: 検索処理

2: 推定処理

3: 確認処理

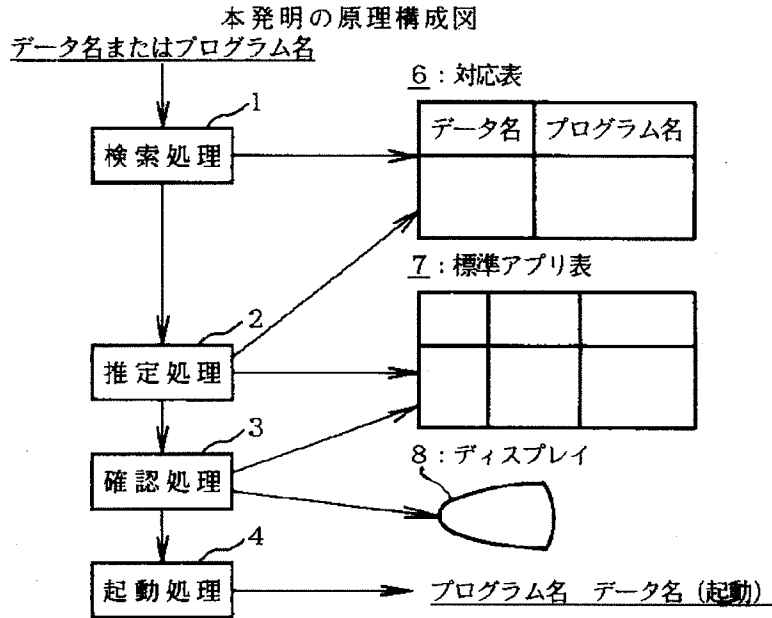
4: 起動処理

10 6: 対応表

7: 標準アプリ表

8: ディスプレイ

【図1】



【図2】

本発明の対応表例  
6: 対応表

データ名	プログラム名
文書A	ワープロB
文書C	ワープロB
⋮	⋮

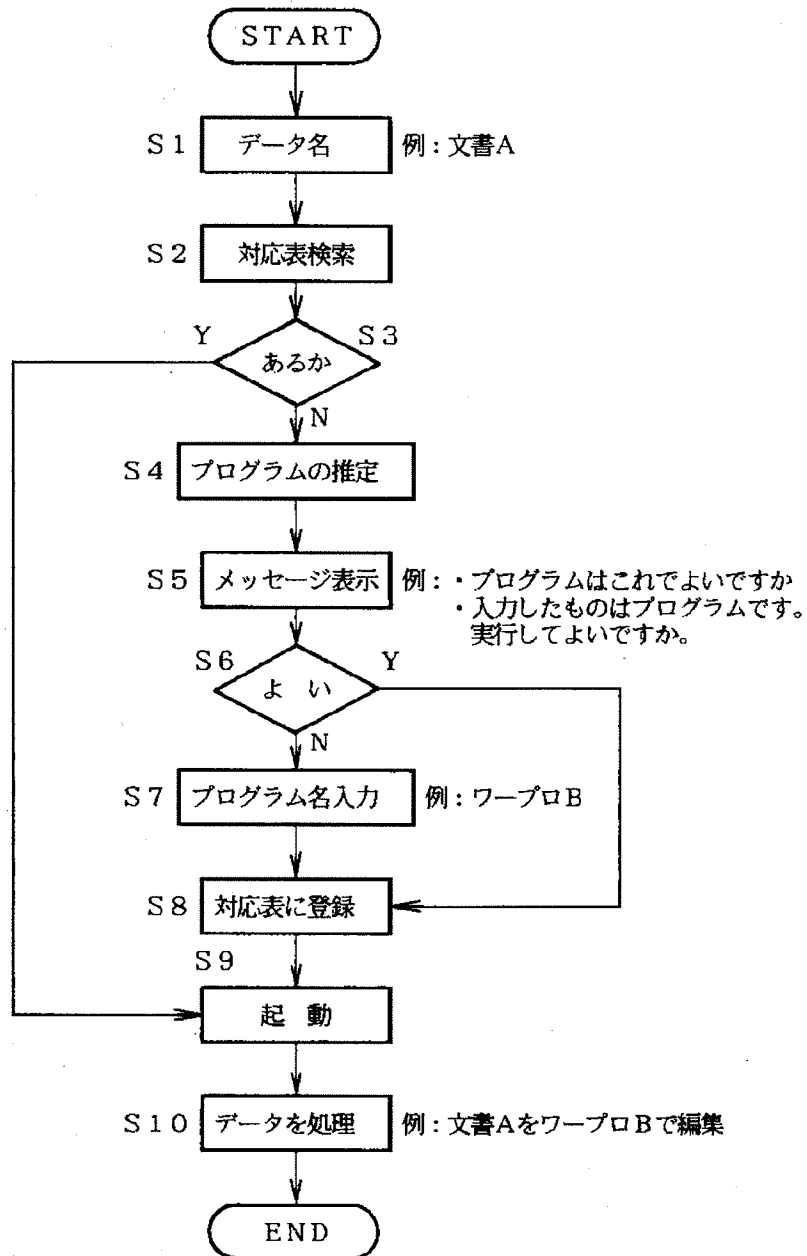
【図3】

本発明の標準アプリ表例  
7: 標準アプリ表

データの種別	プログラム名	データの特徴
テキスト	ワープロB	
図形	ICAD	
⋮	⋮	⋮
データベース	DB2	特定のヘッダ等

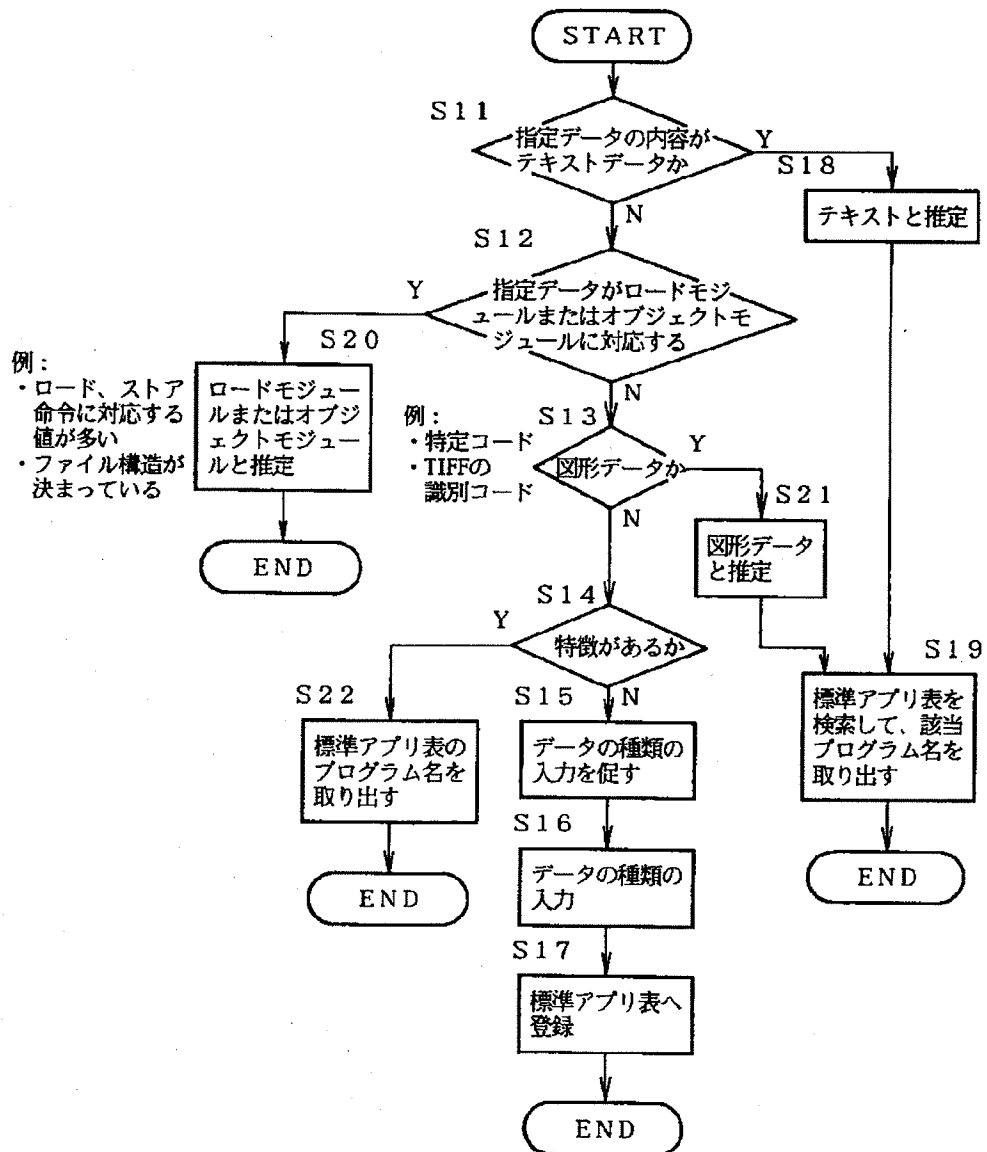
【図4】

## 本発明の動作説明フローチャート



【図 5】

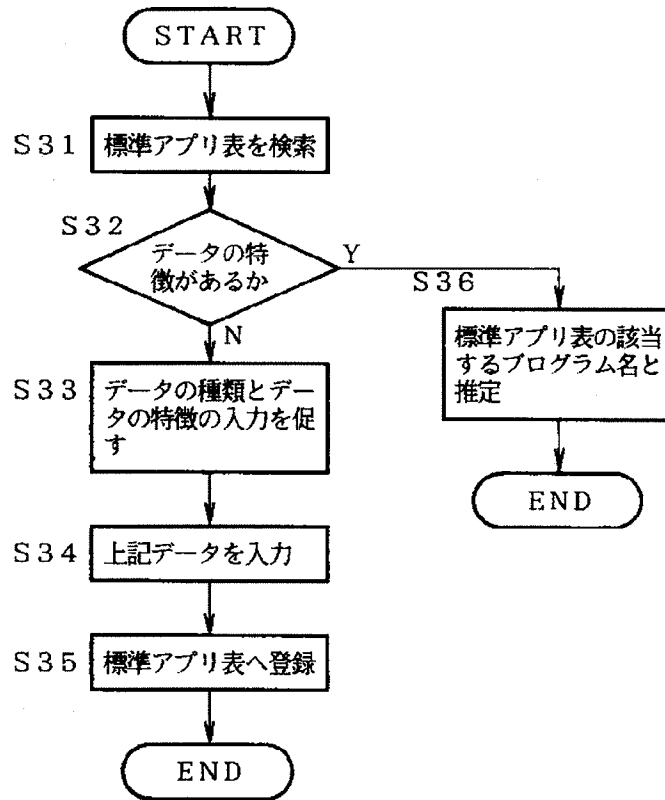
## 本発明の推定説明フローチャート (その1)





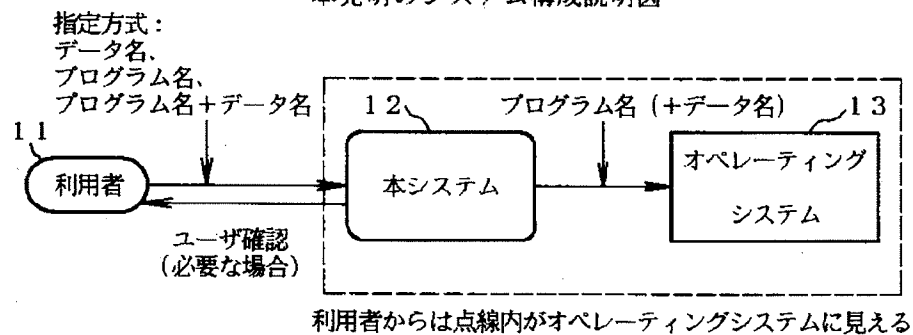
【図 6】

本発明の推定説明フローチャート (その2)



【図 7】

本発明のシステム構成説明図



# METHOD FOR STARTING PROCESSING

**Publication number:** JP6230844 (A)

**Publication date:** 1994-08-19

**Inventor(s):** MOCHIZUKI SHINICHI

**Applicant(s):** FUJITSU SOOSHIARU SCI RABORATO

**Classification:**

- international: **G06F1/00; G06F9/06; G06F9/445; G06F12/00; G06F1/00; G06F9/06; G06F9/445; G06F12/00;**  
(IPC1-7): G06F1/00; G06F9/06

- European:

**Application number:** JP19930016677 19930204

**Priority number(s):** JP19930016677 19930204

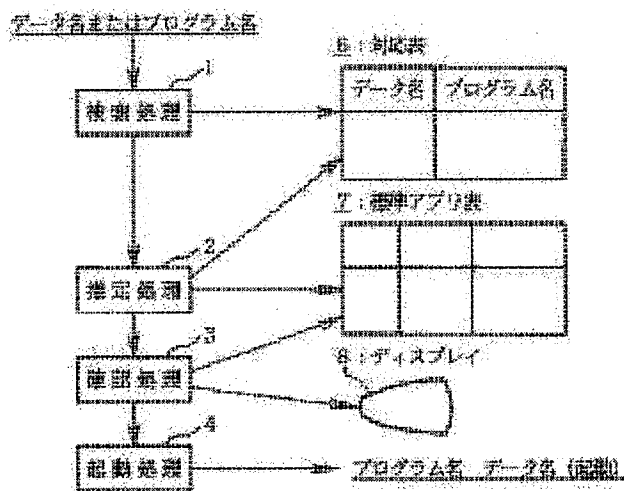
**Also published as:**

JP2695590 (B2)

## Abstract of JP 6230844 (A)

**PURPOSE:** To simply start a program only by specifying a data name by preparing a correspondence table and a reference application table, executing retrieval correspondingly to the specification of a data name and displaying a program name and an estimated program name to start a program with respect to a processing starting method for starting processing.

**CONSTITUTION:** The correspondence table 6 for correspondingly registering a data name and a program name is prepared and retrieved correspondingly to the specification of a data name, its corresponding program name is displayed and a program is started based upon the displayed program name and the specified data name correspondingly to an execution instruction.



**\* NOTICES \***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**CLAIMS**

---

**[Claim(s)]**

[Claim 1]Have a conversion table (6) which matches and registers a data name and a program name in a process starting method which starts processing, and it corresponds to specification of a data name, A process starting method constituting so that a program may be started by a data name which displayed a program name which searches the above-mentioned conversion table (6) and corresponds, and was specified as the displayed program name concerned corresponding to execution instruction.

[Claim 2]Have the following, when a program name which searches and corresponds finds the above-mentioned conversion table (6) corresponding to specification of a data name, display the program name, and on the other hand, when not found, the above-mentioned standard — an application — a table (7) being searched, and a program name corresponding to a kind of data of the specified data name concerned being taken out, and it displaying, and, A process starting method constituting so that a program may be started by a data name specified as the program name concerned corresponding to execution instruction and which starts processing.  
A conversion table (6) which matches and registers a data name and a program name.  
a standard which matches with a kind of data and registers a program name — an application — a table (7).

[Claim 3]Have the following, when a program name which searches and corresponds finds the above-mentioned conversion table (6) corresponding to specification of a data name, display the program name, and on the other hand, when not found, the above-mentioned standard — an application — a table (7) being searched, and a program name corresponding to the feature of data of the specified data name concerned being taken out, and it displaying, and, A process starting method constituting so that a program may be started by a data name specified as the program name concerned corresponding to execution instruction and which starts processing.  
A conversion table (6) which matches and registers a data name and a program name.  
a standard which matches with the feature of data and registers a program name — an application — a table (7).

[Claim 4]When execution instruction of other program names is inputted and carried out corresponding to a display of the above-mentioned program name, From claim 1 statement constituting so that a data name and other program names by which execution instruction was carried out which were specified may be matched and it may register with the above-mentioned conversion table (6) to the process starting method according to claim 3

[Claim 5]the above-mentioned standard — an application, when a table (7) is searched and neither [ concerned / which were specified ] a kind of data of a data name nor the feature is found (or — accepting necessity), corresponding to at least one input of a kind of data of the data name concerned, and the feature of data — the above-mentioned standard — an application — claim 3 statement constituting so that it may match with a program name and may register with a table (7), and the process starting method according to claim 4.

---

[Translation done.]

\* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention relates to the process starting method which starts processing.

[0002]

[Description of the Prior Art]When data was specified and a this compatible program was started conventionally, one of the following methods was adopted.

[0003](1) Embed the information on a program into data. Thereby, a user's specification of data will start a program [ / based on the program information currently embedded in it ].

[0004](2) Add the program information which uses it for the name of data. Thereby, if a user specifies data, the program information added to the name of data will be taken out, and a this compatible program will be started.

[0005]

[Problem(s) to be Solved by the Invention]There is complicated nature which embeds the information on a program into data according to the conventional method of (1) mentioned above, According to the method of (2), there was complicated nature which adds the information on a program to the name of data, it cared about the kind of data to the user, or operation became complicated, and all had the problem that a program could not be started by easy operation.

[0006]in order that this invention may solve these problems -- a conversion table and a standard -- an application -- a table, [ provide and ] It aims at displaying the program name which searches these and corresponds corresponding to specification of a data name, and the presumed program name, starting a program, and starting a program simply only by specifying a data name.

[0007]

[Means for Solving the Problem]Drawing 1 shows a principle lineblock diagram of this invention. In drawing 1, the conversion table 6 matches and registers a data name and a program name.

[0008]a standard -- an application -- Table 7 is matched with a kind of data, and the feature of data, and registers a program name.

[0009]

[Function]This invention displays the program name which searches the conversion table 6 and corresponds corresponding to specification of a data name, and he is trying to start a program by the data name specified as the displayed program name concerned corresponding to execution instruction, as shown in drawing 1.

[0010]When the program name is displayed when the program name which searches and corresponds finds the conversion table 6 corresponding to specification of a data name, and not found on the other hand, a standard -- an application -- Table 7 is searched, and he takes out and displays the program name corresponding to the kind of data of the specified data name concerned, and is trying to start a program by the data name specified as the program name concerned corresponding to execution instruction

[0011]When the program name is displayed when the program name which searches and

corresponds finds the conversion table 6 corresponding to specification of a data name, and not found on the other hand, a standard — an application — Table 7 is searched, and he takes out and displays the program name corresponding to the feature of the data of the specified data name concerned, and is trying to start a program by the data name specified as the program name concerned corresponding to execution instruction

[0012]Under the present circumstances, when execution instruction of other program names is inputted and carried out corresponding to the display of a program name, he matches the data name and other program names by which execution instruction was carried out which were specified, and is trying to register with the conversion table 6.

[0013]a standard — an application, when Table 7 is searched and neither [ concerned / which were specified ] the kind of data of a data name nor the feature is found (or — accepting necessity), corresponding to at least one input of the kind of data of the data name concerned, and the feature of data — a standard — an application — he matches with a program name and is trying to register with Table 7

[0014]therefore, the conversion table 6 and a standard — an application — by forming Table 7, displaying the program name which searches these and corresponds corresponding to specification of a data name, and the presumed program name, and starting a program corresponding to execution instruction, It becomes possible to start a program simply only by specifying a data name.

[0015]

[Example]Next, the composition and operation of the example of this invention are explained to details one by one using drawing 7 from drawing 1.

[0016]Drawing 1 shows the principle lineblock diagram of this invention. that the retrieval processing 1 finds the PUROGURA name or data name which searches the conversion table 6 and corresponds based on the specified data name or program name in drawing 1 \*\*\*\* — etc. — it carries out.

[0017]the case where the estimation processing 2 is not able to find the program name or data name which corresponds even if the retrieval processing 1 searches the conversion table 6 — a standard — an application — Table 7, [ search and ] finding the program name corresponding to the features (for example, an object module, a load module, etc. with a specific instruction code) of the data of a data name in finding the program name corresponding to the kinds (for example, a text, a figure, etc.) of data of a data name \*\*\*\* \*\*\*\* — etc. — it carries out.

[0018]The confirming processing 3 displays the program name corresponding to the data name found by the retrieval processing 1 or the estimation processing 2, and the data name corresponding to a program name on the display 8, or, corresponding to this display — receiving the confirmation input of O.K. from a user and making execution of the starting processing 4 start \*\*\*\* — etc. — it carries out.

[0019]The starting processing 4 starts a program based on the data name and program name which the user checked and carried out execution instruction. The conversion table 6 matches and registers a data name and a program name. For example, it registers with a data name (for example, the document A) and a program name (word processor B) (refer to drawing 2).

[0020]a standard — an application — Table 7 is matched with the kind of data, and the feature of data, and registers a program name. For example, the feature (for example, there are many kanji codes) of the kind (for example, text) of data, a program name (for example, the word processor B), and data is registered (refer to drawing 3).

[0021]Drawing 2 shows the example of a conversion table of this invention. This conversion table 6 is matched with a data name, and registers a program name, for example, registers it as follows [ a graphic display ].

Data name Program name document A word processor B document C the word processor B, thus by registering, When a user specifies the document A, this conversion table 6 is searched, the program name of the word processor B is taken out, a program is started based on these data names "document A" and a program name "word processor B", and the document A is processed (edit). under the present circumstances — the case where it does not register with the conversion table 6 — the standard of drawing 2 — an application — Table 7 is searched

and it presumes and displays, and when the execution instruction from a user occurs, the data name and program name at this time are registered into the conversion table 6 concerned. Thereby, it learns and the result which was presumed automatically and checked is registered into the conversion table 6 one by one. When not registering with the conversion table 6, it matches with the data name which had the program name inputted from the screen specified, and may be made to register.

[0022]drawing 3 --- the standard of this invention --- an application --- the example of a table is shown. this standard --- an application --- Table 7 matches and registers the kind of data, the feature of data, and a program name, for example, registers them as follows [ a graphic display ].

[0023]

Kind of data Program name The feature text of data There are many word processor B kana and kanji codes. Figure ICAD Database DB2 by [ , such as a specific header, ] registering in this way, For example, when a user specifies a text name and the text name is not found in the conversion table 6, the program B compatible with the kind (or the feature of a text) of the text concerned can be taken out, presumed and displayed. When a user looks at this program name presumed and displayed, it is judged to be suitable and execution instruction is carried out, a text name and a program name are registered into the conversion table 6 mentioned above. a user seeing, judging that it is not suitable on the other hand, and waiting for the input of the new program name corresponding to the text name (data name) concerned --- a standard --- an application --- it registers with Table 7 (or correction).

[0024]Next, according to an order shown in the flow chart of drawing 4, operation of the composition of drawing 3 is explained in detail from drawing 1. In drawing 4, S1 specifies a data name. For example, a user specifies "the document A."

[0025]S2 searches a conversion table. As for this, the retrieval processing 1 of drawing 1 searches the program name corresponding to the data name (for example, the "document A") specified by S1 from the conversion table 6.

[0026]S3 distinguishes whether there is any program name which S2 searches and corresponds. Since the program name corresponding to the data name specified by S1 searched and found the conversion table 6 in YES, it starts by S9 and data is processed by S10. On the other hand, since in NO searched the conversion table 6 and it was not found, it progresses to S4.

[0027]S4 presumes a program. this follows the flow chart of drawing 5 mentioned later --- a standard --- an application --- Table 7 is searched, and the program name corresponding to the kind of data of a data name or the feature of data specified by S1 is found and presumed.

[0028]S5 displays the message of the program name found and presumed by S4. For example, as indicated on right-hand side, may this be sufficient as - program?

[0029]- What was inputted is a program. May I perform?

S6 distinguishes whether it inputted, when the user was good corresponding to the message indicator of S5. The data name and program name which were presumed and checked in YES are registered into the conversion table 6 by S8, and it progresses to S9. On the other hand, in NO, the presumed program name is not suitable, or since it was not able to presume, the input of a program name is demanded from a user by S7. And the data name specified as this inputted program name by S8 is registered into the conversion table 6, and it progresses to S9.

[0030]S9 starts a program. This starts a program based on the data name and program name (S4 the standard the application the program name which searched and took out the conversion table 6 by S2, the program name which searched and found Table 7 and the user inputted by presumption, the checked program name, or S7) which were specified by S1.

[0031]S10 processes data. This edits the document A with the word processor B, for example. By the above, when the program name which searches and corresponds finds the conversion table 6 corresponding to specification of a data name, this program is started and data is processed. When searched the conversion table 6 and it is not found, a standard --- an application --- search Table 7, take out, presume and display the program name corresponding to the kind of data of a data name, or the feature of data, a user checks, a data name and a program name are registered into the conversion table 6 at the time of O.K., and this program is started and data is processed, a user --- NO or a standard --- an application --- when not found

from Table 7, a data name and a program name are registered into the conversion table 6 corresponding to the input of a program name, and this program is started and data is processed. By these composition and learning functions, a user becomes possible [ starting a program easily by easy operation of specifying a data name ].

[0032] Drawing 5 shows the presumed explanation flow chart (the 1) of this invention. This is a detail flowchart of presumption of the program of S4 of drawing 4. In drawing 5, the contents of the data distinguish S11 in text data. Since the specified data name was not registered into the conversion table 6, the contents of the data (the data) of the the-data name concerned distinguish this in text data. The interval of text data of - newline character is less than 300 bytes here, - And since the values (line feed, a screen clearance, etc.) to 0-2f<sub>16</sub> have the character in which it is few and there are many values (kanji code) more than 80<sub>16</sub> as - and "a" to "z", distinguish whether the character concerned exists in the contents of the data. presuming text data by S18 in YES -- S19 -- a standard -- an application -- Table 7 is searched, an applicable program name is taken out, and presumption is ended. On the other hand, in NO of S11, it progresses S12.

[0033] S12 distinguishes whether the data correspond to a load module or an object module. Here, since - file structure with many values (code) corresponding to [ load / - / and ] store instruction has the character in which it is decided that it will be a specific form, a load module or an object module distinguishes whether the character concerned exists in the contents of the data. By S20, in YES, a load module or an object module is presumed, and it ends in it. On the other hand, in NO of S12, it progresses S13.

[0034] S13 is distinguished in graphic data. Here, since graphic data have the character to have an identification code of - specific code and TIFF, they distinguish whether the character concerned exists in the contents of the data. presuming graphic data by S21 in YES -- S19 -- a standard -- an application -- Table 7 is searched, an applicable program name is taken out, and presumption is ended. On the other hand, in NO of S13, it progresses S14.

[0035] S14 distinguishes whether there is any feature of data. Here, the feature of data is the feature that a specific header etc. exist, for example in the data, and distinguishes whether the feature of the data concerned exists. the case of YES -- S22 -- a standard -- an application -- the program name of Table 7 is taken out and presumption is ended. On the other hand, in NO, it progresses S15.

[0036] The input of the kind of data is urged to S15. S16, a user inputs the kind of data. S17 -- a standard -- an application -- it registers to a table. S17 is NO in all of S11 to S14 from these [ S15 ] -- a standard -- an application, since it became clear that it cannot search from Table 7, making a user input the kinds (for example, a text, a figure, a database, etc.) of the data corresponding to the message to which the input of the kind of data is urged, and making the program name which processes specification DETA input -- the kind and program name of data -- a standard -- an application -- it registers with Table 7 and prepares for next time.

[0037] the time of not existing in the conversion table 6 by the above corresponding to specification of a data name -- a standard -- an application -- Table 7 is searched, and the program name corresponding to the kinds (a text, a figure, etc.) of data of the specified data name and the features (specific header etc.) of data is taken out and presumed. a thing applicable when there are not a kind of data and the feature of data -- a standard -- an application -- it matches with a program name and registers with Table 7. It becomes possible to enable this to presume the program name corresponding to the specified data name according to the kind and the feature of data of the data name concerned, and to learn.

[0038] Drawing 6 shows the presumed explanation flow chart (the 2) of this invention. this -- a basis [ feature / of data ] -- a standard -- an application -- it is a procedure when searching Table 7. in drawing 6 -- S31 -- a standard -- an application -- a table is searched. the feature of the data of a data name that this was specified -- a standard -- an application -- Table 7 is searched.

[0039] S32 distinguishes whether there is any feature of data. the case of YES -- the feature of data -- a standard -- an application -- since Table 7 was suited -- S36 -- a standard -- an



application — the program name to which Table 7 corresponds is presumed. on the other hand — the case of NO — the feature of data — a standard — an application — since there was nothing to Table 7, it progresses to S33.

[0040]The input of the kind of data and the feature of a program is urged to S33. S34, a user inputs the above-mentioned data. S35 — a standard — an application — it registers to a table. the feature of the data of a data name that S35 was specified by NO of S32 from these [ S33 ] — a standard — an application, since it did not register with Table 7, a user inputting the kind of data, and the feature of data and matching the kind of these data, the feature of data, and also the program name to start corresponding to having demanded the input of the kind of data, and the feature of data from the user, — a standard — an application — it registers to Table 7.

[0041]the feature of the data of the data name specified by the above — a standard — an application — a program name applicable when registering with Table 7 being presumed, and, demanding the kind of data, the feature of data, and the input of a program name from a user, when not registered, and matching what was inputted — a standard — an application — it registers with Table 7. the program name corresponding to the kind of next time to data, and the feature of data by this — a standard — an application — since it registers with Table 7, it becomes possible to presume immediately.

[0042]Drawing 7 shows the system configuration explanatory view of this invention. This combines this system 12 of the composition of drawing 1 with the operating system of an electronic computer. Here, this system 12 may be formed in the input side of the existing operating system 13 like a graphic display, and is good also as a part of operating system 13. In any case, from a user, the inside of a dotted line seems to have indicated in the lower berth in an operating system. It becomes possible to start a program and to perform processing of data according to the specification method of either a data name, a program name or a program name + data name, without being conscious of this system 12.

[0043]In drawing 7, as a specification method, the user 11 specifies either — data name, a program name and a program name + data name, inputs it into this system 12, starts a program, and directs processing of data.

[0044]This system 12 has the composition of drawing 1, and carries out a starting request by a program name + data name at the operating system 13 also to one specification of a data name, a program name, and a program name + data name.

[0045]The operating system 13 is a system which carries out generalization control of the whole electronic computer, starts a program and makes data process here corresponding to the starting request which specified the program name + data name (for example, starting the word processor B mentioned already the document A edit).

[0046]

[Effect of the Invention]As explained above, when according to this invention the program name is displayed when the program name which searches and corresponds finds the conversion table 6 corresponding to specification of a data name and it is not found on the other hand, a standard — an application — Table 7 being searched, and the program name corresponding to the kind (or the feature of data) of data of the specified data name concerned being taken out, and it presuming, displaying, and, Start a program based on the data name specified as the program name concerned corresponding to execution instruction, or, registering the data name specified as the presumed program name into the conversion table 6 \*\*\*\* — the kind, the feature, and program name of data — a standard — an application, since the composition registered into Table 7 is adopted, a program can be easily started only by specifying a data name — both — the conversion table 6 and a standard — an application — the data name, the program name, the kind of data, and the feature of data by which gave the learning function to Table 7 and execution instruction was carried out to it can be registered. By these, it is (1). Even if there is no function special to an operating system, this system can be added, a program corresponding only by specifying data can be started, and processing can be performed.

[0047](2) Also about the data name which exists before introducing this invention, by the estimation processing 2, a program corresponding only by specifying data can be started and

---

[Translation done.]

**\* NOTICES \***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

**DESCRIPTION OF DRAWINGS**

---

[Brief Description of the Drawings]

[Drawing 1]It is a principle lineblock diagram of this invention.

[Drawing 2]It is an example of a conversion table of this invention.

[Drawing 3]It is a standard application front example of this invention.

[Drawing 4]It is an explanation flow chart of this invention of operation.

[Drawing 5]It is a presumed explanation flow chart (the 1) of this invention.

[Drawing 6]It is a presumed explanation flow chart (the 2) of this invention.

[Drawing 7]It is a system configuration explanatory view of this invention.

[Description of Notations]

1: Retrieval processing

2: Estimation processing

3: Confirming processing

4: Starting processing

6: Conversion table

7: a standard --- an application --- a table

8: Display

---

[Translation done.]

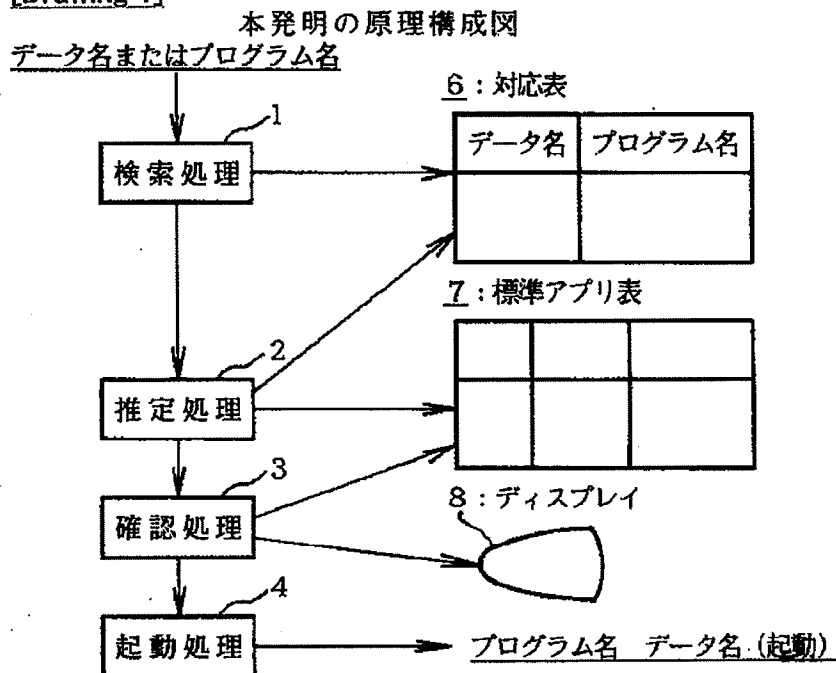
## \* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

## DRAWINGS

[Drawing 1]



[Drawing 2]

本発明の対応表例  
6: 対応表

データ名	プログラム名
文書A	ワープロB
文書C	ワープロB
⋮	⋮

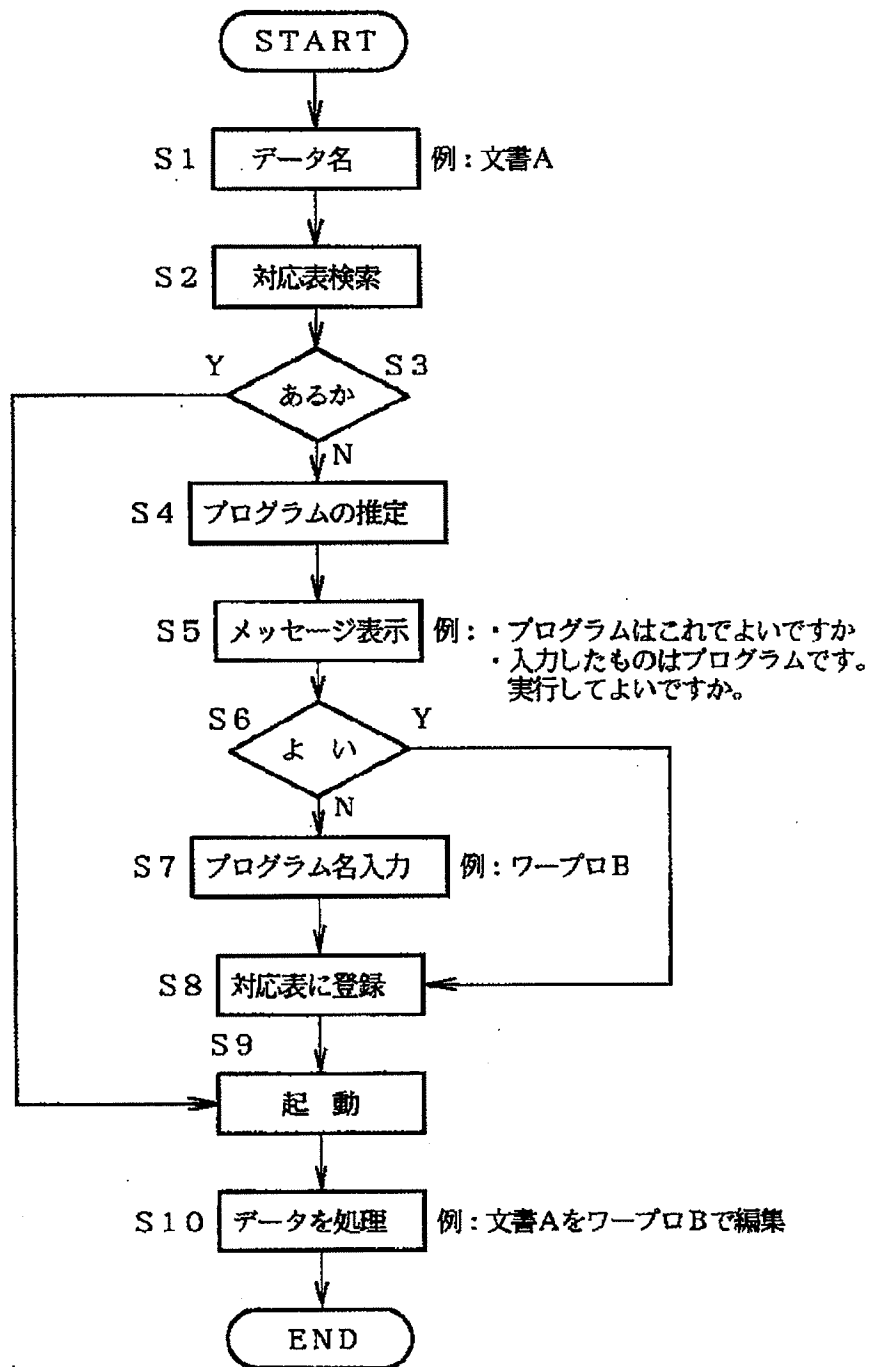
[Drawing 3]

本発明の標準アプリ表例  
7: 標準アプリ表

データの種別	プログラム名	データの特徴
テキスト	ワープロB	
図形	ICAD	
⋮	⋮	⋮
データベース	DB2	特定のヘッダ等

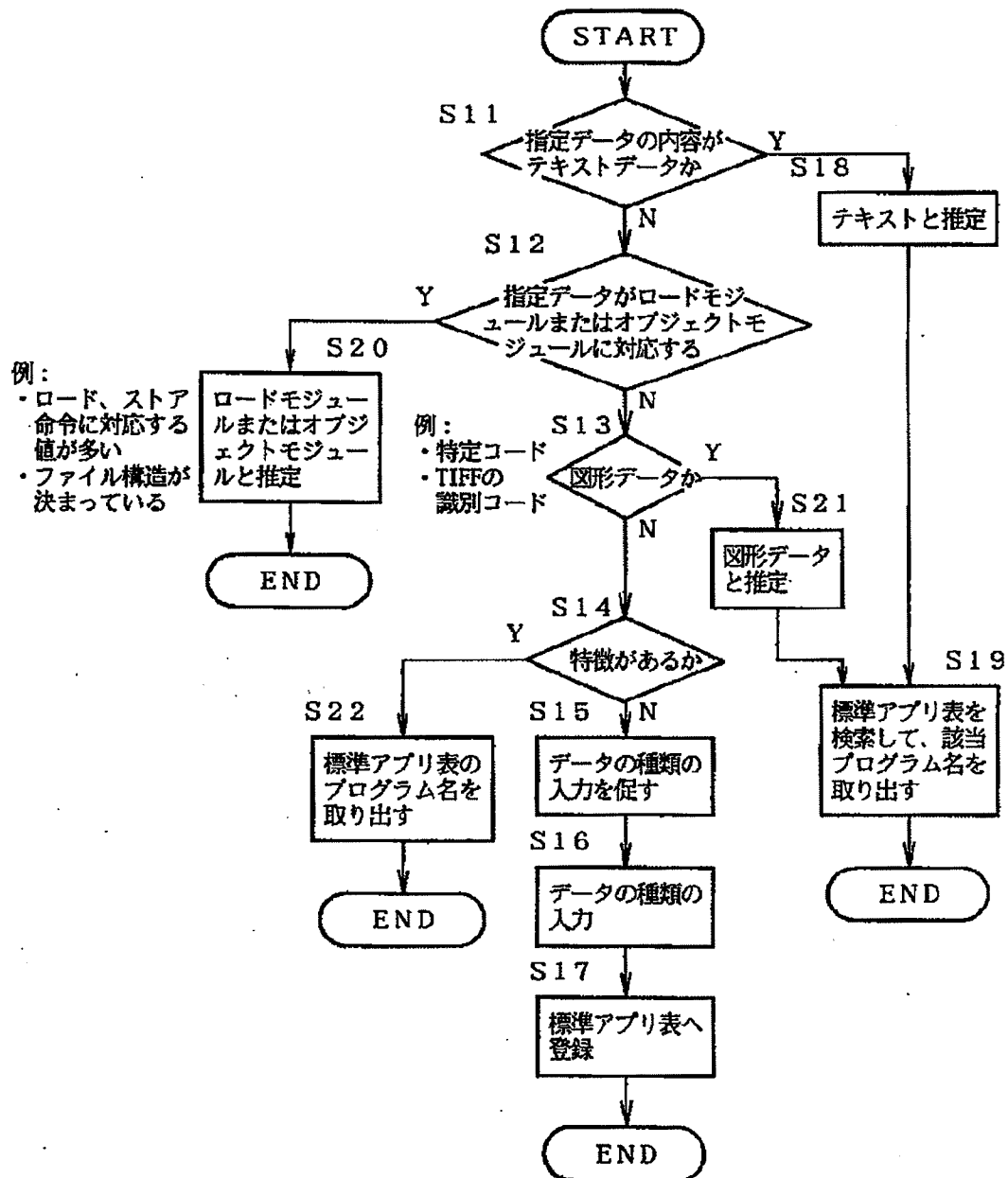
[Drawing 4]

## 本発明の動作説明フローチャート



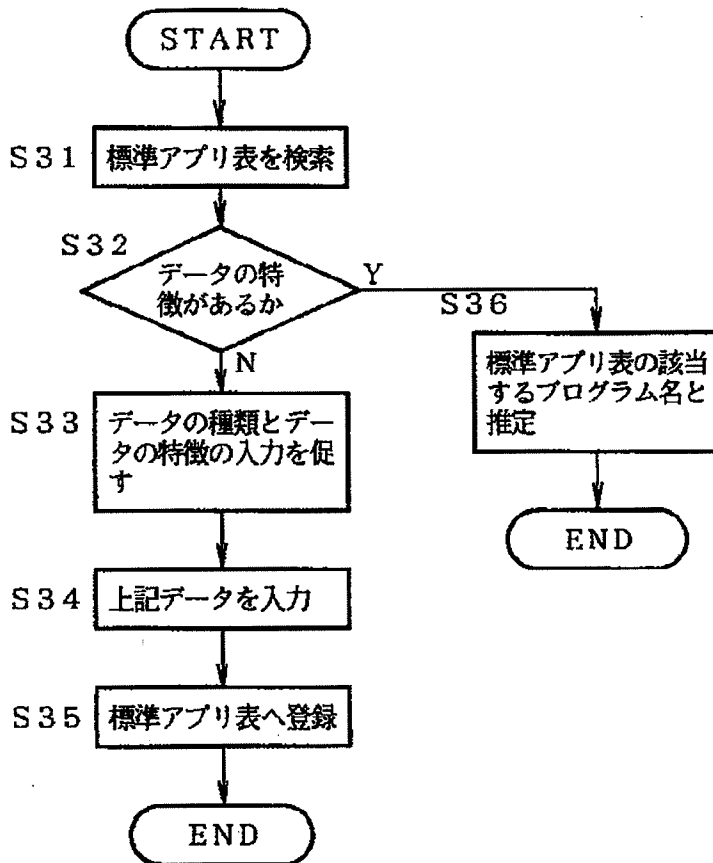
[Drawing 5]

本発明の推定説明フローチャート (その1)



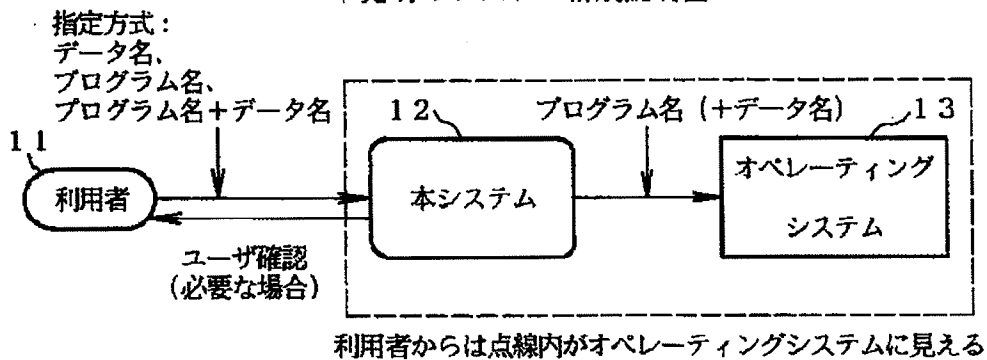
[Drawing 6]

## 本発明の推定説明フローチャート (その2)



## [Drawing 7]

## 本発明のシステム構成説明図



[Translation done.]